



HAL
open science

GÉNÉRATION AUTOMATIQUE DE DIAGRAMMES FORMELS PAR UN ALGORITHME COGNITIF MODULAIRE : ÉTUDE PRÉLIMINAIRE

Joséphine Calandra, Jean-Marc Chouvel, Myriam Desainte-Catherine, Pierre
Hanna

► **To cite this version:**

Joséphine Calandra, Jean-Marc Chouvel, Myriam Desainte-Catherine, Pierre Hanna. GÉNÉRATION AUTOMATIQUE DE DIAGRAMMES FORMELS PAR UN ALGORITHME COGNITIF MODULAIRE : ÉTUDE PRÉLIMINAIRE. Journées d'Informatique Musicale, JIM2020, Oct 2020, Strasbourg, France. hal-03362923

HAL Id: hal-03362923

<https://hal.science/hal-03362923>

Submitted on 2 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

GÉNÉRATION AUTOMATIQUE DE DIAGRAMMES FORMELS PAR UN ALGORITHME COGNITIF MODULAIRE : ÉTUDE PRÉLIMINAIRE

Joséphine Calandra

IReMus, LaBRI

josephine.calandra@labri.fr

Jean-Marc Chauvel

IReMus

jeanmarc.chouvel@free.fr

Myriam Desainte-Catherine

Pierre Hanna

LaBRI

myriam@labri.fr

hanna@labri.fr

RÉSUMÉ

Nous abordons dans cet article une étude de cas servant d'illustration et de preuve de concept concernant la question de la systématisation de l'analyse paradigmatique par la création automatique de diagrammes formels, en appliquant ce que nous appelons un algorithme cognitif. Par cognitif, nous entendons la simulation hypothétique d'une machine d'écoute et non du système réel de l'écoute. Un diagramme formel est une représentation des matériaux constitutifs de l'œuvre musicale par ordre d'apparition, en fonction du temps. Nous présentons ici un système modulaire permettant de produire un diagramme formel en prenant un signal monophonique en entrée, synthétisé à partir d'un fichier MIDI. L'objectif de ce système à long terme est de générer une structure composée de superposition de diagrammes formels dont les matériaux des niveaux supérieurs constituent un groupement de matériaux du niveau inférieur direct et ce à partir d'œuvres musicales sans limites de genres.

1. INTRODUCTION

L'analyse de la musique vise à une meilleure compréhension de la musique. Souvent l'analyse considère la musique comme un objet spatialisé dans une trace, en particulier sous forme de partition. Il convient pourtant de restituer ce phénomène dans le temps, en tant que processus cognitif. Les sciences cognitives se sont beaucoup intéressées au phénomène musical, et il apparaît déterminant pour l'analyse de mieux comprendre et de mieux représenter l'intentionnalité temporelle au cœur des œuvres.

Avec l'émergence de la théorie de l'information (Shannon [17]) et de la sémiologie (Nattiez [14]), les recherches de François Delalande [7] ont montré l'influence du sujet dans la compréhension du musical. D'autre part, les compositeurs du vingtième siècle, comme John Cage [19], ont proposé d'étendre la musique à l'ensemble du sonore.

Devant la difficulté à définir la musique de manière immanente, on a par ailleurs essayé de mieux comprendre les effets de la musique, car depuis très longtemps, on considère la musique comme un des plus puissants vecteurs de l'émotion. Gisèle Brelet [3] soutient même que la musique suscite en nous « l'être que nous devrions être ».

Le psychologue John Sloboda [18] a ainsi réalisé une expérience au cours de laquelle 83 personnes ont recensé des œuvres leur ayant fait ressentir des émotions particulières, en indiquant les éléments musicaux à l'origine de ces émotions. Par une analyse très précise de la partition, ils ont pu déduire que certaines formes musicales spécifiques donnaient lieu à des émotions spécifiques. Par exemple les harmonies inattendues provoquaient des frissons tandis que les appoggiatures provoquaient les pleurs.

Puisque la musique semble être le reflet du système cognitif, on a depuis les travaux d'Otto Laske [10] proposé de modéliser un système d'analyse musicale qui s'apparente à une « machine à écouter », selon le titre du livre d'Émile Leipp [11]. Si le sonagramme a profondément modifié la compréhension du monde sonore, il manque encore à l'analyse musicale un « morphoscope » [13] opérationnel, permettant de visualiser l'évolution temporelle de l'œuvre en tenant compte des impératifs cognitifs.

Cet article s'attache plus spécifiquement à implémenter la notion de diagramme formel [4]. On appelle dans cet article mémoire matériau le dictionnaire correspondant à l'œuvre musicale analysée. Ces diagrammes représentent l'ensemble des occurrences des matériaux d'une œuvre, classés dans leur ordre d'apparition, en fonction du temps. C'est une première étape avant d'aborder la notion de structure, et d'explicitier l'ensemble des niveaux intermédiaires qui pourraient être la note, le fragment musical, la phrase, etc.

Une telle analyse peut être effectuée à la main, mais cette démarche est d'une part fastidieuse, et elle implique d'autre part des choix subjectifs concernant en particulier les catégories de matériaux pris en compte par le musicologue. On souhaite donc automatiser cette analyse, à la fois pour simplifier cette tâche, favoriser ainsi l'émergence d'une analyse comparée, mais aussi pour donner une consistance concrète aux pré-connaissances du musicologue. Dans la version finale, on envisage également un monitoring de l'algorithme automatisé, afin de mieux comprendre quels processus cognitifs sont impliqués par les stratégies de composition déployées par les compositeurs.

On présente dans le présent article un système modulaire permettant à partir d'un fichier audio de générer un diagramme formel au niveau 0, à partir d'un échan-

tillonnage élémentaire du signal où l'on impose comme matériau une fenêtre (*frame*) de taille donnée. Dans la section 2, nous décrivons les travaux existants favorisant l'analyse automatisée de la forme ou de la structure des œuvres musicales. Dans la section suivante, nous définissons l'algorithme cognitif, montrons des exemples obtenus manuellement et à partir de fichiers MIDI puis précisons le cadre d'étude de l'algorithme. Nous décrivons ensuite l'architecture modulaire du système permettant d'obtenir un diagramme formel à partir d'un fichier audio monophonique obtenu par synthèse d'un fichier MIDI. Nous détaillons l'objectif de chacun des modules, leur état actuel et leurs perspectives. Enfin, nous décrivons dans la dernière section les expériences permettant d'analyser les résultats obtenus et nous proposons un algorithme permettant d'automatiser la validation des diagrammes.

2. ÉTAT DE L'ART

Les sémiologues cherchent à comprendre d'une part ce qui fait entité, autrement dit comment segmente-t-on, et d'autre part ce qui fait identité, à savoir comment compare-t-on. En linguistique, Ferdinand de Saussure [8] développe les notions duales de paradigme et de syntagme qui correspondent respectivement aux éléments de répétitions et à la répartition temporelle de ces éléments. Seulement, la musique se différencie de la linguistique en cela qu'elle n'a pas de mots pour la construire et est propice à une variété de styles innombrables. Il est donc nécessaire de définir des critères plus généraux de segmentation et de similarité.

Une autre question est de savoir s'il est possible d'automatiser l'analyse paradigmatique. En informatique musicale, de nombreux outils existent déjà pour permettre l'analyse musicale. De manière générale, la stratégie employée est de calculer des descripteurs audio afin de faciliter le repérage des ruptures d'homogénéité et de répétition. Cela permet de s'attaquer à une écriture de plus haut niveau qui permettrait de classer sous une même catégorie deux éléments sensiblement différents. On peut citer parmi les descripteurs majoritairement utilisés : les *Mel Frequency Cepstral Coefficients* pour le timbre, les *chromas* qui permettent d'associer au signal l'énergie moyenne de la gamme, et pour le rythme, on analyse les émissions des notes, c'est-à-dire les attaques, et les sons percussifs par l'analyse de l'évolution du spectre d'intensité sonore. On peut de manière générale calculer de nombreux descripteurs permettant de donner par exemple les valeurs de la pulsation, du tempo, de la métrique, de l'harmonie et combiner ces différentes approches dans une analyse multi-strate [16].

Benjamin Martin [12] propose un algorithme pour analyser la structure musicale à partir de l'audio. Il reprend un algorithme de comparaison de chaînes de caractères dans les séquences d'ADN pour éviter lors de la phase de classification d'effectuer toutes les comparaisons possibles et ainsi gagner en temps de calcul. Cela permet par ailleurs l'utilisation de descripteurs robustes à l'altération de mo-

tifs et à la transposition. Il propose également un modèle hiérarchique du système en représentant l'œuvre par un arbre qu'il produit en recherchant de manière descendante les sous-structures des structures détectées. Néanmoins, Benjamin Martin ne s'attaque pas aux problématiques de la temporalité ni de l'effet sur le cognitif. Par ailleurs, il se restreint aux styles musicaux conventionnels, qui permettent une certaine répétition dans la structure de l'œuvre.

La thèse de Gabriel Sargent [16] propose une estimation de la structure d'un morceau de musique à partir de l'audio par une approche différente, en s'inspirant des notions linguistiques de paradigme et de syntagme pour définir ce qu'il appelle une structure sémiotique : il crée un principe méthodologique d'analyse structurale à partir de l'étude interne des blocs, grâce au modèle système-contraste [2]. Il effectue également une analyse paradigmatique qui étudie les relations entre les blocs d'une même classe et les oppositions entre blocs de classes différentes, afin de permettre l'étiquetage des segments, et une analyse syntagmatique qui étudie les blocs vis-à-vis de leurs voisins. Dans son étude, il n'y a pas de hiérarchisation et il choisit lui-même la taille des segments auquel il souhaite se référer. Gabriel Sargent se limite également aux styles musicaux conventionnels.

Antescofo, développé par l'IRCAM [6] est un logiciel de suivi de partitions synchronisées au jeu du musicien. Il analyse des caractéristiques par toutes sortes de descripteurs et agit en fonction d'événements inscrits sur la partition. Il est également équipé d'un système qui permet de prévoir le tempo du musicien afin de s'adapter à celui-ci. Il modélise ainsi un système d'écoute qui prend en compte l'évolution temporelle de l'œuvre et est capable de reconnaître un élément qui a déjà surgi par le passé. *Ossia Score* [1], développé au sein du SCRIME pose les problèmes de hiérarchisation temporelle en intégrant dynamiquement le temps dans l'exécution de la partition par le logiciel.

Ce qui se rapproche le plus de notre système reste néanmoins la matrice d'autosimilarité [9] qui consiste en une matrice symétrique dont l'ordonnée comme l'abscisse représente le déroulement temporel du morceau et où chaque valeur $s(i, j)$ correspond à la distance entre les descripteurs calculés à l'instant i et à l'instant j de celui-ci. La différence majeure avec l'algorithme cognitif est que cette matrice est calculée entièrement sur l'ensemble de l'œuvre sans considérer le déroulement temporel de celle-ci. Le procédé est donc différent mais on peut néanmoins retrouver le diagramme formel par simplification de la matrice d'autosimilarité, en supprimant les doublons des matériaux.

Il existe donc de nombreuses méthodes d'analyse musicales automatisées, mais soit elles procèdent à l'analyse de l'œuvre directement sur son ensemble sans considération du déroulement temporel de l'œuvre, soit le matériau d'analyse a une taille imposée et on ne trouve pas de construction ascendante de l'analyse de l'œuvre. On souhaite donc proposer un système qui permet une

construction d'une part soucieuse de l'ordre chronologique d'apparition des matériaux, d'autre part dont le matériau d'analyse est construit de manière hiérarchique par strate reposant sur la construction des niveaux inférieurs.

On s'intéresse dans cet article uniquement de manière horizontale à la considération temporelle du déroulement de l'œuvre, à un unique niveau, le niveau de la fenêtre du signal et sur des fichiers audio générés en synthétisant des fichiers MIDI. Nous présentons donc ici une étude de cas basée principalement sur le *Geisslerlied Maria muoter reinû maît* modélisé en MIDI et synthétisé au piano, qui, de part son étude, nous permet de valider ou non les différents concepts mis en place dans l'implémentation du système. Un *Geisslerlied* est un chant allemand populaire de flagellation monophonique, celui-ci datant de 1349. Nous avons choisi cette œuvre car il s'agit du premier exemple pris par Nicolas Ruwet [15] pour effectuer une analyse paradigmatique, reprise par un des auteurs en illustration du concept de diagramme formel [5]. Étant monophonique, cette œuvre semblait par ailleurs assez simple pour une première approche.

3. L'ALGORITHME COGNITIF

L'algorithme cognitif [5] est le procédé permettant de produire un diagramme formel. Il s'agit de la représentation des matériaux d'une œuvre en fonction du temps. Cette représentation permet d'apporter une projection formelle de l'œuvre musicale propice à l'interprétation musicologique. On notera que la notion de diagramme formel, introduite précédemment par un des auteurs [4], précise l'analyse paradigmatique en forçant la lecture dans le sens des axes déterminés par le diagramme : le lecteur est invité à suivre un procédé de lecture précis qui induit l'aspect formel de l'œuvre. Notamment, la notion de front de découverte, qui correspond au tracé concret des nouveaux matériaux sur le diagramme, illustre formellement la stratégie du compositeur.

3.1. Présentation de l'algorithme cognitif

L'algorithme cognitif [5] permet de donner une représentation formelle d'une œuvre musicale. Il se présente de la manière suivante :

```
pour chaque fragment entendu :
  comparaison avec les fragments
  précédents ;
  s'il est différent :
    inscription en mémoire
    « matériau » ;
  dans tous les cas: inscription en
  mémoire « forme » ;
```

On représente donc l'information de l'œuvre en tant que matériau, par ordre d'apparition du matériau, en fonction du temps. L'algorithme que nous utilisons ici n'est en réalité qu'une partie de l'algorithme cognitif dans son

entièreté : dans un second temps cet algorithme est réitéré dans un mécanisme plus complexe pour une approche hiérarchique de la construction des matériaux par groupement des matériaux de niveau inférieur. On l'appelle cognitif car les fragments de niveau inférieur sont regroupés ou non à l'instant t pour former des fragments de niveau supérieur. Ceci est permis par une étude de segmentation et de similarité mise en œuvre par des processus de reconnaissance paradigmatique et syntagmatique en lien avec les connaissances propres à l'œuvre qui sont mises à jour régulièrement par mémorisation des nouveaux fragments acquis par apparition chronologique de niveau n et $n + 1$. Ceci est également permis grâce aux connaissances externes à l'œuvre, la protension d'objets hypothétiques et éventuellement leur réalisation par critères de segmentation. Nous ne nous penchons pas sur cette question dans l'article et nous limitons ici à l'analyse de premier niveau de l'œuvre, au niveau du signal, où l'on boucle une partie de l'algorithme sur un seul niveau d'analyse.

On qualifie cet algorithme de cognitif dans le sens d'une hypothèse de simulation d'une machine d'écoute et non pas comme modélisation du système réel de l'écoute. On pourra en revanche parler de modélisation dans le sens où l'on cherchera à représenter différentes conduites d'écoute au sens de François Delalande [7], c'est-à-dire différentes approches de l'œuvre, en fonction du paramétrage du système. En effet, il n'y a pas un seul diagramme formel possible pour une œuvre musicale, mais autant de diagrammes possibles qu'il y a d'approches d'écoute d'une œuvre. Le matériau en question n'est donc pas spécialement défini. Manuellement, le musicologue choisit donc à quelle échelle et sur quel critère il souhaite analyser l'œuvre.

3.2. Applications triviales de l'algorithme

On peut générer un diagramme formel à partir de différentes représentations. À partir d'une partition ou d'un fichier MIDI, la segmentation à l'échelle de la note est déjà donnée par la durée de la note ou le *onset* indiqué dans la représentation. L'étude de similarité est également triviale car le label de la note est donné, par son emplacement sur la portée ou par sa dénomination dans le fichier MIDI. On peut donc obtenir sans difficulté le diagramme formel d'une œuvre musicale.

Un exemple historique est l'analyse du *Geisslerlied* intitulé *Maria muoter reinû maît*, qui donne, en appliquant l'algorithme cognitif avec la note pour fragment, à partir de la partition en figure 1, le diagramme formel en figure 2.



Figure 1. Début de partition du *Geisslerlied*.

On a également cherché à représenter des diagrammes formels à partir d'un fichier MIDI. On a pour cela im-

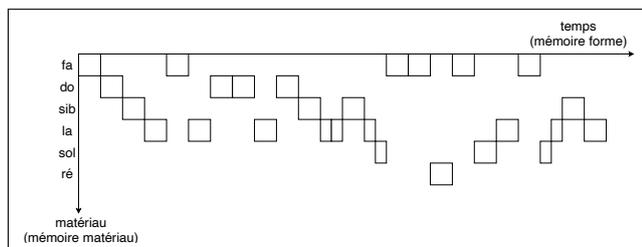


Figure 2. Diagramme formel du *Geisslerlied*.

plémenté l'algorithme cognitif en récupérant les informations adéquates du fichier MIDI avec la bibliothèque Python music21¹. On étudie les 3 premières mesures et deux temps du prélude de la *Suite I pour violoncelle seul* de J.-S. Bach, en appliquant l'algorithme cognitif avec la note comme fragment. À partir du fichier MIDI au tempo 60 bpm représenté en figure 3, on obtient le diagramme formel donné en figure 4.

```
<MidiFile 1 tracks
<MidiTrack 0 -- 232 events
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent SEQUENCE_TRACK_NAME, t=None, track=0, channel=None, data=b'\x00'>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent TIME_SIGNATURE, t=None, track=0, channel=None, data=b'\x04\x02\x08'>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent TIME_SIGNATURE, t=None, track=0, channel=None, data=b'\x04\x02\x08'>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent NOTE_ON, t=None, track=0, channel=1, pitch=43, velocity=100>
<MidiEvent DeltaTime, t=24, track=0, channel=None>
<MidiEvent NOTE_OFF, t=None, track=0, channel=1, pitch=43, velocity=64>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent NOTE_ON, t=None, track=0, channel=1, pitch=50, velocity=100>
<MidiEvent DeltaTime, t=24, track=0, channel=None>
<MidiEvent NOTE_OFF, t=None, track=0, channel=1, pitch=50, velocity=64>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
...
<MidiEvent NOTE_ON, t=None, track=0, channel=1, pitch=55, velocity=100>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent NOTE_OFF, t=None, track=0, channel=1, pitch=59, velocity=64>
<MidiEvent DeltaTime, t=24, track=0, channel=None>
<MidiEvent NOTE_OFF, t=None, track=0, channel=1, pitch=55, velocity=64>
<MidiEvent DeltaTime, t=0, track=0, channel=None>
<MidiEvent END_OF_TRACK, t=None, track=0, channel=None, data=b''>
>
```

Figure 3. Extrait du fichier MIDI du début du prélude de la *Suite I pour violoncelle seul* de J.-S. Bach.

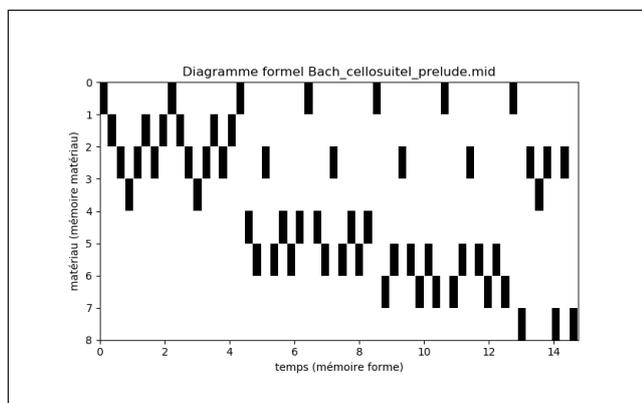


Figure 4. Diagramme formel du début du prélude de la *Suite I pour violoncelle seul* de J.-S. Bach.

3.3. Interprétations du diagramme formel

On rappelle que l'intérêt de l'analyse par diagramme formel ne se porte pas uniquement à l'échelle de la note. L'analyse peut également être opérée sur des matériaux plus subtils ou plus larges qui peuvent être plus pertinents à d'autres échelles, mais également sur d'autres dimen-

¹<http://web.mit.edu/music21/>, accédé le 19 octobre 2020

sions sonores (par exemple l'amplitude ou le rythme) et dont la représentation s'éloigne de celle du *piano roll*. Il s'agit par ailleurs d'un principal défi au niveau de l'implémentation de l'algorithme que de déterminer automatiquement quelles seront les échelles et les critères pertinents de segmentation de matériau. Néanmoins, même à l'échelle de la note, cette représentation se différencie de la représentation de la partition et du *piano roll* (figure 5) par la représentation temporelle de l'apparition de nouveau matériaux, qui met en exergue la forme de l'œuvre dans son évolution temporelle et le front de découverte, mettant ainsi en lumière les différentes stratégies employées par les compositeurs.

C'est donc cet algorithme de génération automatique de diagramme formel à un premier niveau que l'on souhaite modéliser dans un premier temps. À la différence de la méthodologie employée ici, on procédera à une analyse à partir de fichiers audio.

Par ailleurs, la génération de l'algorithme cognitif pour objectif premier d'aider les musicologues en proposant une représentation des œuvres musicales qu'ils seront à même d'interpréter par eux-même, nous n'avons donc pas la prétention dans ce système d'interpréter automatiquement musicalement les diagrammes obtenus. En revanche, une analyse du second ordre du fonctionnement même du système de génération des diagrammes nous permettra sans doute à terme d'apporter d'autres informations concernant l'assimilation par une machine d'écoute des œuvres musicales dans le temps.

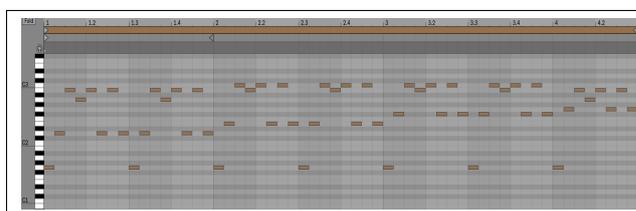


Figure 5. *Piano roll* du début du prélude de la suite I pour violoncelle seul de J.-S. Bach. On trouve en ordonnées les notes en écriture anglaise, en abscisse les mesures.

4. ARCHITECTURE DU SYSTÈME À PARTIR DU NIVEAU AUDIO

Toute la difficulté de la segmentation et de l'étude de similarité se joue lorsque l'on cherche à obtenir le diagramme formel à partir de l'audio. En effet, il faudra alors trouver leurs critères à l'aide de calculs de descripteurs et d'analyses sur l'énergie et l'amplitude du signal.

Nous avons donc cherché à implémenter un système (figure 6) qui permet de prendre un fichier audio et des paramètres en entrée, et de fournir un diagramme formel en sortie. Le système est constitué de quatre modules, de manière à pouvoir remplacer n'importe quelle partie indépendamment des autres. Nous avons pour le moment implémenté de manière primaire chacune des parties afin de construire l'architecture générale tout en se gardant la possibilité d'améliorer chacune d'entre elles.

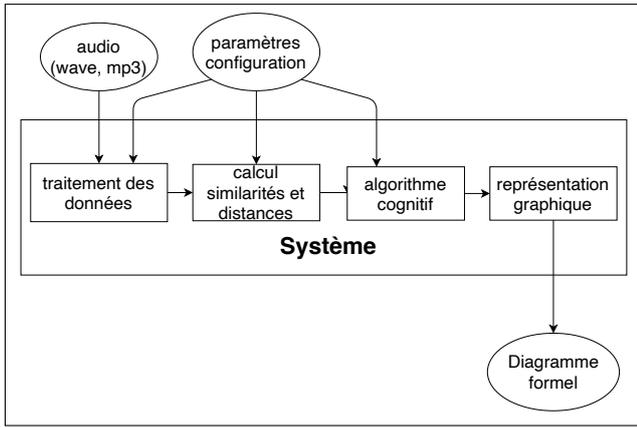


Figure 6. Schéma général du système.

4.1. Configuration

En plus de l'audio, on fournit un paramétrage des différentes variables du système afin d'obtenir la précision souhaitée pour le diagramme. De manière générale, le paramétrage est l'interface qui permettra au musicologue de choisir sous quel angle de vue et selon quelles caractéristiques audio il souhaite produire le diagramme formel.

Ainsi, nous proposons de paramétrer manuellement la longueur de la fenêtre analysée, qui est notre matériau de niveau zéro. De manière générale, la longueur de la fenêtre devra être assez petite pour être inférieure au seuil d'audibilité (on a pris pour référence le temps audible de latence de 0,3 ms), mais il doit également être assez grand pour que le nombre de données analysées dans la fenêtre soit représentatif du signal. Pour une fréquence d'échantillonnage du signal de 44 100 Hertz, on favorise donc une longueur de fenêtre de 1024 valeurs, ce qui correspond à une durée de fenêtre d'environ 23 millisecondes.

On propose également de choisir manuellement le seuil de similarité. Le seuil de similarité permettra de déterminer lors des calculs de distance à partir de quelle valeur de similarité l'élément actuel est considéré comme identique ou non à un élément précédent. C'est donc ce qui détermine la précision en terme de nombre de matériaux du système. Afin de pouvoir comparer nos diagrammes à ceux qui sont produits manuellement ou à partir de fichiers MIDI, on choisit par tâtonnement des seuils qui nous permettent d'obtenir des similarités telles que l'ensemble des fenêtres soient similaires par leur hauteur de note.

Dans le cas où le diagramme est généré à partir du calcul de *mel frequency cepstral coefficients* (*mfcc*), on permet aussi de choisir le nombre de *mfcc* que l'on souhaite calculer pour chaque fenêtre.

À terme, on pourrait procéder à un calcul optimal des paramètres par exemple par calcul de l'entropie [20].

4.2. Traitement des données

Le module de traitement des données permet de passer de l'audio à l'ensemble des descripteurs qui seront traités pour l'étude des similarités. Pour le moment nous avons calculé les spectres à partir de la Transformée de Fourier Rapide et les *mfcc*, l'amplitude *RMS* (*Root Mean Square*)

du signal sur les fenêtres pour obtenir des informations sur le volume, ainsi que les différentielles en fréquence pour obtenir des informations sur les intervalles. Nous n'avons cependant pas encore déterminé les critères de similarités pour les deux derniers descripteurs.

On pourrait proposer à l'avenir également une génération automatique de descripteurs par *machine learning*. Cela pourrait néanmoins nuire à la transparence totale de système, nécessaire pour la compréhension des processus mis en œuvre dans la création des diagrammes formels.

4.3. Similarités et distances

On calcule dans ce module différentes distances. De la même manière que pour le calcul des distances dans une matrice d'autosimilarité, nous avons dans un premier temps calculé la distance euclidienne D_e . La distance euclidienne correspond à la distance au plus court entre chacune des *mfcc* terme à terme des deux fenêtres étudiées. On l'applique donc sur l'ensemble des *mfcc* de chaque fenêtre où l'on note d_n l'ensemble des N *mfcc* de la fenêtre n . On le traduit par la formule (1). Pour obtenir une valeur de similarité s , on calcule $s = 1 - D_e$.

$$D_e(d_n, d_m) = \sqrt{\sum_{i=1}^N (d_n(i) - d_m(i))^2} \quad (1)$$

L'intérêt des *mfcc* est qu'ils recouvrent en peu de valeurs et de manière assez puissante l'information de l'enveloppe spectrale du signal.

Néanmoins, le calcul des *mfcc* fait perdre énormément d'information du signal, et dans l'optique de rester au plus proche du signal complet, Nous avons également calculé la similarité cosinus (2). Ce calcul de similarité est pertinent dans le sens où il permet de calculer l'énergie commune en terme de bandes fréquentielles entre une fenêtre et une autre. On définit \mathbf{A}_n et \mathbf{A}_m fenêtres telles que $\mathbf{A}_n = [a(n, j), j \in N]$ et $\mathbf{A}_m = [a(m, j), j \in N]$ où $a(i, j)$, $i \in (n, m)$ correspond à l'amplitude pour la fréquence j à l'instant i avec N le nombre de fréquences dans la fenêtre.

$$\begin{aligned} S_{\cos}(\mathbf{A}_n, \mathbf{A}_m) &= \cos(\theta) = \frac{\mathbf{A}_n \cdot \mathbf{A}_m}{\|\mathbf{A}_n\| \|\mathbf{A}_m\|} \\ &= \frac{\sum_{i=1}^N \mathbf{A}_n(i) \mathbf{A}_m(i)}{\sqrt{\sum_{i=1}^N \mathbf{A}_n^2} \sqrt{\sum_{i=1}^N \mathbf{A}_m^2}} \end{aligned} \quad (2)$$

Le temps de calcul est néanmoins plus important pour cette méthode.

On pourra à l'avenir tester d'autres distances sur de nouveaux descripteurs, et calculer les distances sur des combinaisons de descripteurs pondérés en fonction de leur importance dans le système. On pourrait également dans ce module avoir une approche classification plutôt que

similarité, c'est-à-dire que l'on labellise chacun des éléments du signal plutôt que d'étudier la similarité entre chacun d'eux.

4.4. L'algorithme cognitif

Ce module constitue l'implémentation de l'algorithme tel que décrit ci-dessus. Le système n'est pas en temps réel, mais il simule l'écoute au fur et à mesure de l'œuvre car on compare chaque descripteur de chaque fenêtre par ordre temporel avec tous ceux des fenêtres précédentes jusqu'à trouver une fenêtre similaire. Si la valeur de comparaison entre les deux fenêtres est supérieure au seuil donné en paramètre, alors la nouvelle fenêtre est notée comme étant le même matériau que l'ancienne fenêtre. Si aucune fenêtre précédente n'est reconnue comme similaire à l'actuelle, alors un nouveau matériau est créé.

On pourra à l'avenir améliorer les performances de cet algorithme en évitant de comparer chaque fenêtre à toutes les précédentes, par exemple en implémentant le *Variable Markov Oracle* par Shlomo Dubnov [20].

4.5. Représentation graphique

On crée donc une matrice dont on ajoute une ligne pour chaque nouveau matériau trouvé, et dont les colonnes sont mises à jour au fur et à mesure que l'on avance dans l'écoute de l'œuvre musicale. La valeur dans la matrice est de 0 dans la case $m(i, j)$ si la fenêtre à la date i correspond au matériau j et 1 sinon.

Pour représenter les différents niveaux hiérarchiques de l'œuvre, on envisage une représentation 3D où la troisième dimension correspondrait au diagrammes formels des différents niveaux. Il faut néanmoins réfléchir à une superposition adéquate au niveau supérieur du regroupement des matériaux de niveau inférieur.

5. VALIDATIONS EXPÉRIMENTALES

On se situe ici très spécifiquement dans le cadre de mélodies monophoniques mais d'autres cas seront à étudier dans un second temps. Afin de s'abstraire des variabilités de timbre et de dynamique qui peuvent être très importantes dans un enregistrement à partir d'instruments réels, nous avons cherché à voir si l'on réussissait à répondre aux problèmes de structuration et de similarité de l'audio avec des fichiers les plus propres possible : on a donc généré des fichiers audio à partir de synthèse de fichiers MIDI. Cela nous permettra dans un second temps de comparer la correspondance des entités entre le fichier MIDI et audio.

5.1. Analyse d'exemples

Un fichier MIDI du *Geisslerlied* a été généré. On trouve un exemple de référence dans la section 3. Ce fichier a été synthétisé avec un son de piano à queue au tempo 120 bpm.

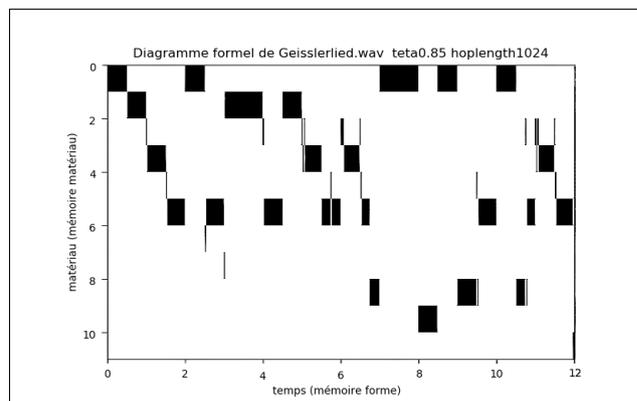


Figure 7. Diagramme formel du *Geisslerlied* en similarité cosinus avec une fenêtre de taille 1024 et un seuil de 0,85.

On obtient un résultat conforme à la référence avec un paramétrage d'une taille de fenêtre de 1024 éléments et un seuil de 0,85.

On constate l'apparition par rapport au diagramme produit manuellement de matériaux transitoires. Effectivement, on est actuellement au niveau de la fenêtre, où sont donc représentés des éléments de transition ainsi que des éléments correspondant à une hauteur de note. Le niveau hiérarchique de la note est donc très certainement le niveau direct supérieur.

On compare maintenant les résultats en faisant varier les différents paramètres. En modifiant la taille de la fenêtre, on constate que la précision de l'étude de la similarité est modifiée. En effet, lorsque l'on diminue la taille de la fenêtre (figure 8), celle-ci est trop petite pour obtenir des résultats cohérents à l'échelle auditive. Il est possible d'une part que la fenêtre soit assez petite pour que l'évolution de la dynamique de la note soit détectée, ce qui n'est pas forcément pertinent. Notamment, l'attaque d'une note pour un son percussif tel que le piano peut être perçue comme très différente de la suite de la note. D'autre part, c'est toute l'étude de similarité qui pourra être faussée car deux attaques percussives de notes différentes peuvent être considérées comme similaires car ayant des fréquences similaires à fortes amplitudes à une telle échelle de temps. Lorsque la fenêtre est plus grande (figure 9), l'étude de la similarité est moins précise : on obtient des éléments transitoires beaucoup plus gros avec une fenêtre de taille 2048 par rapport à une fenêtre de taille 1024.

En faisant varier le seuil de similarité, on observe que lorsque l'on diminue ce seuil (figure 10), on perd en précision et en nombre de matériau, simulant ainsi une écoute beaucoup plus générale de l'œuvre, tandis que si l'on augmente ce seuil (figure 11), on aura une précision beaucoup plus fine de l'écoute, ou chaque transition voir modification de dynamique du timbre dans la note sera repéré, augmentant ainsi le nombre de matériaux obtenus.

On peut regrouper les fenêtres de transitions avec les fenêtres suivantes correspondantes à une même note, d'une part car l'ordre de grandeur des transitions par rapport aux regroupement de fenêtres d'une même hauteur est très inférieure (moins d'un dixième de la note),

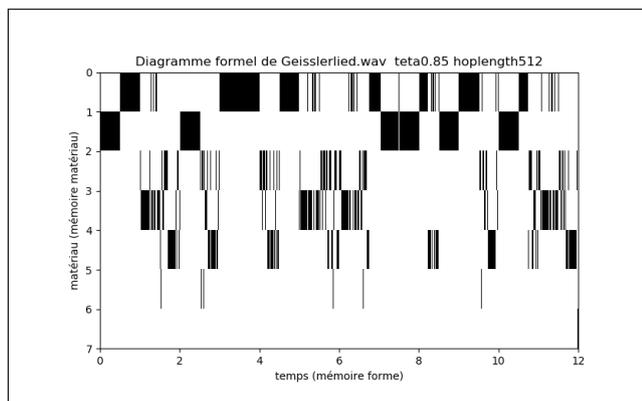


Figure 8. Diagramme formel du *Geisslerlied* en similarité cosinus avec une fenêtre de taille 512 et un seuil de 0,85.

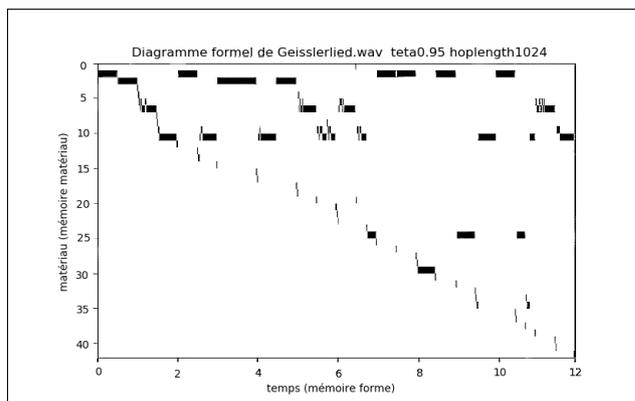


Figure 11. Diagramme formel du *Geisslerlied* en similarité cosinus avec une fenêtre de taille 1024 et un seuil de 0,95.

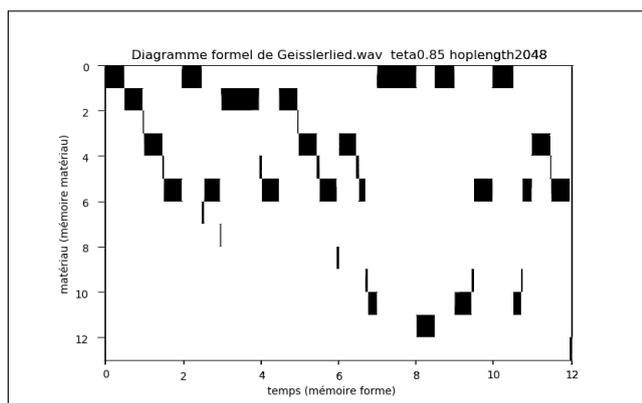


Figure 9. Diagramme formel du *Geisslerlied* en similarité cosinus avec une fenêtre de taille 2048 et un seuil de 0,85.

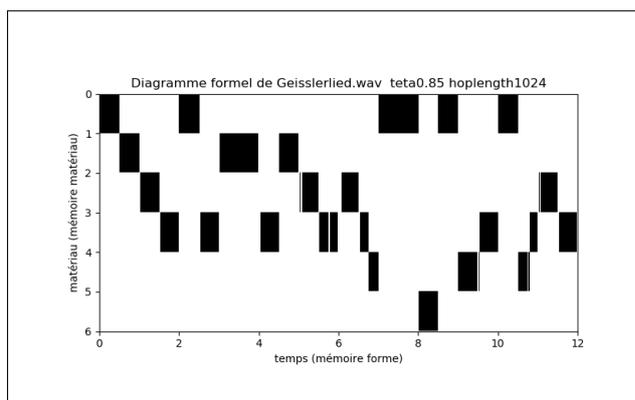


Figure 12. Diagramme formel de niveau 1 du *Geisslerlied* en similarité cosinus avec une fenêtre de taille 1024 et un seuil de 0,85.

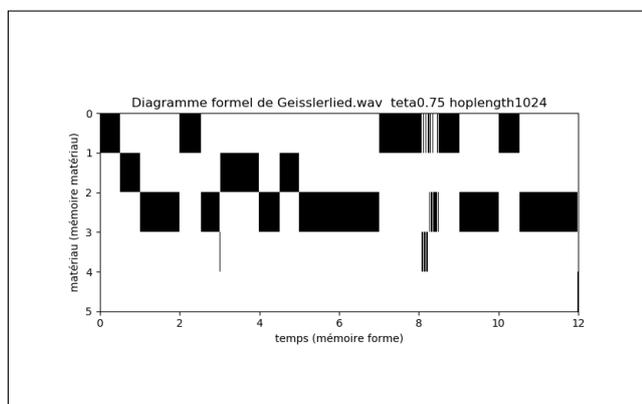


Figure 10. Diagramme formel du *Geisslerlied* en similarité cosinus avec une fenêtre de taille 1024 et un seuil de 0,75.

et que l'on ne souhaite pas considérer simultanément deux ordres de grandeur différents. D'autre part, les éléments transitoires ne sont pas autonomes et sont liés aux éléments qui les précèdent et les succèdent, il est donc légitime de les regrouper avec ceux-ci. On présente en figure 12 un diagramme formel de niveau 1, où l'on a gommé les éléments transitoires, qui ne sont plus pertinents à ce niveau. On pourra ainsi comparer ces diagrammes aux diagrammes formels obtenus à partir du MIDI.

Nous avons également modifié le fichier MIDI du *Geisslerlied* de deux manières. Nous avons d'abord dupliqué le morceau en deux, une première partie à l'octave actuelle et une seconde partie à l'octave inférieure. Pour le même audio et la même taille de fenêtre, on obtient avec un seuil inférieur un même matériau pour deux octaves différentes (octave non détectée) : la similitude obtenue avec un seuil inférieur se justifie certainement par un spectre très proche entre deux mêmes notes sur des octaves différentes. Pour un seuil plus élevé, on observe un nouveau matériau (octave détectée), auquel cas le diagramme n'indique pas qu'il s'agit *a priori* de la même note. La deuxième modification a consisté à reprendre des parties du *Geisslerlied* à un volume différent et on remarque que les nuances ne sont pas repérées, ce qui semble cohérent avec la normalisation lors du calcul de la distance cosinus.

5.2. Vers une validation automatique

On cherchera ensuite à valider automatiquement les diagrammes formels de niveau 1, c'est-à-dire sans considération des transitions.

Nous proposons pour cela de comparer les diagrammes obtenus automatiquement à partir du fichier MIDI avec

ceux obtenus à partir de l'audio de ce même fichier MIDI. Nous souhaitons pour cela utiliser notamment une banque de fichiers MIDI séquencée par David J. Grossman de l'ensemble des suites pour violoncelle seul de J.-S. Bach².

Nous avons pensé dans un premier temps calculer la différence de recouvrement entre chacune des matrices en calculant la somme des différences terme à terme des valeurs pixels des deux matrices, et en divisant le tout par le nombre de matériaux multiplié par le nombre de fenêtres dans l'œuvre. Seulement, l'apparition ou la disparition d'un matériau entraîne l'apparition ou la disparition d'une ligne et donc le décalage total de l'ensemble des lignes correspondantes aux matériaux apparaissant ultérieurement. La comparaison terme à terme se retrouve alors faussée alors même que les lignes ultérieures sont potentiellement identiques.

Pour pallier le problème de décalage de lignes, nous comparerons pour chaque matrice le front de découverte. Le front de découverte correspond à l'ensemble des couples du numéro d'ordre d'apparition d'un matériau et la date de première apparition de chaque nouveau matériau.

On calculera donc le front de découverte pour chacun des deux diagrammes, à partir de l'audio et du MIDI de référence. La comparaison des deux fronts de découvertes (indice du matériau et date d'apparition de ce nouveau matériau) nous permettra de constater lorsqu'un nouveau matériau n'aura pas été calculé dans un diagramme (pas de nouveau matériau à l'instant t par rapport au diagramme de référence), ou à l'inverse si nous avons un matériau nouveau de trop (le matériau i est à un instant t_c inférieur à l'instant t_r au diagramme de référence). On quantifiera alors dans un premier temps l'efficacité de l'étude de la segmentation en comptant le nombre de matériaux supprimés et le nombre de matériaux ajoutés par rapport au diagramme de référence. On pourra également détecter exactement entre quels matériaux et à quel instant il manque la création d'un nouveau matériau et donc d'une ligne, ou à l'inverse à quel instant un matériau superflu et donc une ligne supplémentaire a été créée.

En gardant uniquement les lignes qui concordent avec le front de découverte des deux diagrammes, on étudiera ensuite l'efficacité de la similarité en faisant la différence matricielle telle que décrite en premier lieu entre les mêmes matériaux des deux diagrammes. On quantifiera ainsi les erreurs par la différence entre chacune des lignes ramenée à la dimension des diagrammes. On pourra également représenter la matrice des différences et détecter précisément à quels instants il y a eu des différences de résultats de similarité entre les deux diagrammes.

6. DISCUSSIONS

Dans le cadre de nos expériences, on force le choix du matériau et de ses paramétrages à l'aide des références que nous possédons déjà. On est en mesure de se deman-

²www.jsbach.net/midi/midi_solo_cello.html,
accédé le 19 octobre 2020

der quelle peut être alors l'optimisation pour une musique sans partition. La notion de structure sera alors fondamentale, puisque c'est celle-là même qui permettra de déterminer le matériau adéquat à chaque niveau hiérarchique.

Les prochaines étapes sont maintenant de traiter des œuvres de tous types et polyphoniques d'une part, d'autre part à implémenter la hiérarchisation du système. Mais avant cela nous devons tester le système de validation proposé en partie 5 sur ce que nous possédons, puis améliorer les modules de traitement audio et de calcul des distances afin de proposer une configuration plus fine du système et proposer des diagrammes formels sous différents angles de similarité, tel que le rythme ou le volume.

7. CONCLUSION

Dans cet article, nous avons cherché à partir de la question de la systématisation de l'analyse paradigmatique à considérer l'ordre chronologique d'apparitions des paradigmes. Ainsi, nous souhaitons représenter automatiquement un diagramme formel d'une œuvre musicale à partir de l'audio associé à l'œuvre. Nous avons donc pour le moment un système subdivisé en modules distincts de configuration, traitement des données, calcul de similarité, calcul de l'algorithme lui-même et représentation graphique. Ce système permet de prendre en entrée un audio monophonique synthétisé à partir d'un fichier MIDI et qui génère en sortie un diagramme formel au niveau hiérarchique le plus bas qui est le niveau du signal. On constate une grande sensibilité du système vis-à-vis de l'audio, ce qui était attendu. La modification de la taille des fenêtres d'analyse implique une précision peut-être trop importante vis-à-vis de la perception lorsque leur taille est diminuée, et implique une diminution de la précision d'analyse lorsque ces fenêtres ont une taille plus importante. Lorsque l'on modifie la valeur du seuil de similarité, les matériaux sont distingués avec moins de précision lorsque le seuil est moins élevé, et plus de précision lorsque le seuil est plus élevé. Avec des réglages adaptés, les résultats obtenus sur le *Geisslerlied* sont encourageants car ils sont visuellement similaires à ceux obtenus lorsque l'on procède à une analyse à partir d'un fichier MIDI à l'échelle déterminée de la segmentation par la note. La segmentation par la note correspond en réalité néanmoins à l'échelle de segmentation de niveau supérieur à l'analyse obtenue pour le fichier audio : il faut en effet segmenter les groupes de fragments à chaque apparition d'un fragment de matériau différent. Enfin, il reste à effectuer une validation automatisée pour confirmer les résultats.

8. RÉFÉRENCES

- [1] Allombert A., Desainte-Catherine M., Assayag G. « Iscore: A system for writing interaction » *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*, Athens, Greece, 2008.

- [2] Bimbot F., Deruty E., Sargent G., Vincent E. « System & contrast : a polymorphous model of the inner organization of structural segments within music pieces », *Music Perception* 33/5 (2016), p. 631-661.
- [3] Brelet, G. *Le temps musical : essai d'une esthétique nouvelle de la musique, 2 Volumes*. Presses Universitaires de France, Paris, 1949.
- [4] Chouvel, J.-M. « Musical form, from a model of hearing to an analytic procedure », *Interface* 22/2 (1993), p. 99-117.
- [5] Chouvel, J.-M. *Analyse Musicale, sémiologie et cognition des formes temporelles*, L'Harmattan, Paris, 2006.
- [6] Cont, A. « ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music », *Proceedings of the International Computer Music Conference*, Belfast, Ireland, 2008.
- [7] Delalande, F. *La musique au-delà des notes*. Presses Universitaires de Rennes, 2019.
- [8] De Saussure, F. *Cours de linguistique générale*. Payot, Paris, 1906.
- [9] Foote J. « Visualizing music and audio using self-similarity », *Proceedings of the ACM International Conference on Multimedia*, Orlando, Florida, USA, 1999.
- [10] Laske, O. E. *Music, Memory and Thought, explorations in Cognitive Musicology*. University Microfilms International, Ann Arbor (MI), 1977.
- [11] Leipp, E. *La Machine à Ecouter, Essai de psycho-acoustique*. Masson, Paris, 1977.
- [12] Martin, B. « Analyse de structures répétitives dans les séquences musicales », thèse de doctorat, sous la dir. de M. Desainte-Catherine, P. Ferraro, Université Sciences et Technologies - Bordeaux I, 2012.
- [13] Mesnage, M. « Morphoscope, a computer system for music analysis », *Interface* 22/2 (1993), p. 119-131.
- [14] Nattiez, J.-J. *Musicologie générale et sémiologie*. Christian Bourgois, Paris, 1987.
- [15] Ruwet N. « Méthodes d'analyse en musicologie », *Revue belge de Musicologie / Belgisch Tijdschrift voor Muziekwetenschap* 20/1 (1966), p. 65-90.
- [16] Sargent, G. « Estimation de la structure de morceaux de musique par analyse multi-critères et contrainte de régularité », thèse de doctorat, sous la dir. de F. Bimbot, E. Vincent, *Université Rennes 1*, 2013.
- [17] Shannon, C. E. « A Mathematical Theory of Communication », *Bell System Technical Journal* 27 (1948), p. 379-423.
- [18] Sloboda, J. « Music Structure and Emotional Response: Some Empirical Findings », *Psychology of Music* 19/2 (1991), p. 110-120.
- [19] Troche, S. « Structure et méthode dans la musique de John Cage : une discipline d'attention », *Nouvelle revue d'esthétique* 9/1 (2012), p. 91-104.
- [20] Wang C., Dubnov S. « The Variable Markov Oracle: Algorithms for Human Gesture Applications » *IEEE Multimedia* 22/4 (2015), p. 52-67.

Texte édité par Corentin Guichaoua.